63-3-1

FTD-TT- 62-1498

TRANSLATION

ALGORITHM FOR SOLVING A TRANSPORT PROBLEM BY APPROXIMATING WITH CONDITIONALLY OPTIMUM PLANS

By

A. L. Lur'ye

FOREIGN TECHNOLOGY DIVISION

AIR FORCE SYSTEMS COMMAND

WRIGHT-PATTERSON AIR FORCE BASE

OHIO

FTD-TT- 62-1498/1

UNEDITED ROUGH DRAFT TRANSLATION

ALGORITHM FOR SOLVING A TRANSPORT PROBLEM BY APPROXIMATING WITH CONDITIONALLY OPTIMUM PLANS

BY: A. L. Lur'ye

English Pages: 13

SOURCE: Russian Book, Vychislitel'naya Matematika, Collection Nr. 7, 1961, pp 151-160

S/558-61-0-7-5-8

THIS TRANSLATION IS A RENDITION OF THE ORIGINAL FOREIGN TEXT WITHOUT ANY ANALYTICAL OR EDITORIAL COMMENT. STATEMENTS OR THEORIES ADVOCATED OR IMPLIED ARE THOSE OF THE SOURCE AND DO NOT NECESSARILY REFLECT THE POSITION OR OPINION OF THE FOREIGN TECHNOLOGY DIVISION.	PREPARED BY: TRANSLATION SERVICES BRANCH FOREIGN TECHNOLOGY DIVISION WP-AFB, OHIO.

Algorithm for solving a Transport Problem by Approximating with Conditionally Optimum Plans

by

A. L. Lur'ye

1. Introduction. Nature of the Method

In literature on linear programming the name of transportation was attached to the following problem [1].

It is necessary to find the values x_{ik} ($i = 1,2,\dots n$; $k=1,2,\dots,m$) satisfying the conditions:

$$\sum_k x_{ik} = a_i, \quad (1)$$

$$\sum_i x_{ik} = b_k, \quad (2)$$

$$x_{ik} \geqslant 0, \quad (3)$$

$$\sum_{i,k} c_{ik}x_{ik} = \min. \quad (4)$$

where a_i, b_k, c_{ik} - given actual numbers, whereby $a_i > 0$, $b_k > 0$, $\sum a_i = \sum b_k$.

One of the ways of solving the problem of linear programming , proposed by L.V. Kantorovich in 1939 [3], is the approximation with "conditionally optimum plans " [4]

1. This problem was formulated and solved in application to planning transportations over a given transportation network using the method of "circular dependences " in Soviet planned literature in 1930 [1-2]

In this report is discussed the corresponding algorithm for the transportation problem.[2]

The nature of the proposed method goes briefly down to the following. We select a system of values x_{ik}, satisfying the following conditions[1].

$$x_{i_r}, k_p = 0, \text{ если } c_{i_r k_p} > \min_i c_{ik_p} \quad (5)$$

$$\sum_k x_{ik} \leqslant a_i. \quad (6)$$

$$\sum_i x_{ik} \leqslant b_k. \quad (7)$$

$$\sum_{i,k} x_{ik} = \max. \quad (8)$$

After this we determine the values $N_k = b_k - \sum x_{ik}$. If any given $N_k = 0$, the problem is solved. At $N_k > 0$ we break up the values i into two classes i' and i" in the following manner.

To class i" we refer any i_r, if there is such a value k, at which

$$c_{i_r k} = \min_i c_{ik} \text{ и } N_k > 0. \quad (8a)$$

To this class we add any i_r, if there are such values i" and k, for which $c_{i_r k} = \min_i c_{ik}$ and $x_{i"k} > 0$. Remaining i are referred to class i'.

We determine

$$d = \min_{i',k'} (c_{i'k'} - \min_i c_{ik'}), \quad (8b)$$

where under k' are understood such k values, for which the expression (placed in parenthesis) is positive at given i'.

2. In American literature similar methods of solving a transportation problem have been introduced by Ford and Fulkerson [5] and Munkres [6] under the title of the Hungarian method (the title indicates a relationship between the proposals of American authors with the work of Hungarian mathematicians by the theory of graphs)
1. This system of values x_{ik} will be " the conditionally optimum plan ".

Each c_{ik} is increases to d. Having obtained a changed system of c_{ik} values, we again begin with the above mentioned operations.

At small n and m the employment of the proposed method is very simple[2]. With an increase in n and m upon the selection of x_{ik} values, satisfying conditions (5)-(8) difficulties do originate, for the solution of which it is necessary to develop a special method, and a rapidly growing number of operations necessary for the solution. To reduce this number it is desirable for each stage of solving the problem to possibly more fuller utilizing the results of operations, carried out at preceding stages. In this connection it is necessary to turn to certain complex algorithms.

II. Algorithm of Solving the Problem

Par. 1. Initial Operations and certain definitions

Each line of the matrix $\left[c_{ik}\right]$ will be placed in conformity with the number M_i, and each column - number N_k. It is initially assumed that $M_i = a_i$, $N_k = b_k$. The lines, for which $Mi > 0$, will be called absolutely excessive, columns, for which $N_k > 0$, -non-satisfactory, and the remaining columns - satisfactory.

For each column of k matrix $\left[c_{ik}\right]$ we determine its lowest elements $c_{i_r k} = \min c_{ik}$. Line i_r and column k_p are considered interconnected, provided $c_{i_r k p} = \min c_i k_p$.

We will say that line i_r equips column k_p, if after establishing a certain system of values of variables x_{ik}, there is an inequality $x_{i_r k_p} > 0$.

Certain lines we will further place in conformity with the chain, i.e. in rows of numbers of the form of $i_o k_o; i_1 k_1; \ldots; i_s k_s$, where i_o - number of the line, to which is attributed the chain, k_o - number of the column connected with it, i_1 - number of line, providing column k_o, k_1-number of column, connected with line i_1 and so on. To the line, the chain of which consists of 2(s+1) numbers numbers (s=0,1,2,...) is imparted the rank of s.

We determine the values of variables x_{ik}. We assume that $x_{ik} = 0$, if line i and column k are not interconnected. The values of variables x_{ik}, corresponding to intersections of lines and columns, interconnected with each other, we determine alternately in the following manner. We assume that each x_{ik} is equal to the smallest

of the M_i and N_k numbers and simultaneously we reduce both these numbers to the value of the smallest one of them [1].

If after x_{ik} have been determined, all $N_k = 0$, the problem is solved. When $N_k > 0$ we examine alternately each unsatisfactory column and establish, whether there are interconnected lines, to which no chains have been attributed (the chain could have been attributed when examining the previous column). Having detected such a line, we adopt its chain i_0k_0. The lines which have obtained chains, will be insufficient, and all remaining ones - excessive.

We are going over to par.2, considering thereat the insufficient lines, established as result of preceding operations, as unexamined, i.e. subject to operations, explained in par.2.

Par.2. Review of insufficient lines

We begin the review with one of the nonexamined lines, having the lowest rank, i.e. with one of the lines, to which the shortest chains are attributed. If there are no unexamined lines, we go over to par. 5.

We are establishing whether the examined line (we shall designate it with the number i_1) equips any one column, simultaneously connected with an absolutely excessive line. If such a column has been detected, we go over to par.3. In the absence of column, satisfying the above formulated condition, we establish, whether line i_1 equips any one column, connected with excessive line, which does not appear to be absolutely excessive. We designate the number of such column (if it has been detected) by k_0. We attribute to each excessive line, connected with column k_0, a chain $i_0k_0i_1k_1 \ldots i_sk_s$, where i_0 - number of given excessive line, and $i_1k_1i_2k_2 \ldots \ldots i_sk_s$ - a chain of

Footnote to page 3...2...See brief description of solution of concrete example in report [7] and a more detailed one in report [8]

1. The number of following operations will appear to be, as a rule, smaller, if at definite values x_{ik}, corresponding to intersections of interconnected lines and columns, we adhere to the following order. We determine first the values x_{ik} for these lines, according to which $M_i \geq \sum Nk_i$, where k_i - numbers of columns, connected with line i. Then we find values x_{ik} alternately for each of the remaining lines, beginning with lines, connected with the smallest number of columns. At each line we begin with x_{ik}, corresponding to columns, connected with the smallest number of lines.

the examined line i_1. If line i_o has acquired a chain during previous operations, then this chain is considered as having lost its force. By ch.nging the name, excessive lines, having obtained chains, are changed into insufficient. We turn at first to par 2, assuming that the changed lines as well as line i_1 as unexamined.

If it appears that line i_1 does not feed the columns, connected with any one excessive lines, then we acknowledge line i_1 as examined and we turn to the beginning of par 2.

Par.3.Change in Values of Variables

Assuming that $i_1k_1i_2k_2...i_sk_s$ - chain of insufficient line i_1, feeding column k_o connected with absolutely excessive line i_o. We determine the value $d_s=\min(M_{io}, x_{i_1k_o}, x_{i_2k_1}, ..., x_{i_sk_{s-1}}, N_{ks})$.

We reduce by d_s the values $M_{io}, x_{i_1ho}, x_{i_2k1}, ..., x_{i_sk_{s-1}}, N_{ks}$ and we increase by d_s the values $x_{i_1ko}, x_{i_1k_1}, ..., x_{i_sk_s}$.

If the extreme right element of the series M_{io}, $x_{i_1k_o}, x_{i_2k_1}, ..., x_{i_sk_{s-1}}$, N_{k_s}, equal in d_s, was found to be M_{io}, then line i_o is acquiring a chain $i_ok_oi_1k_1...i_sk_s$ and by renaming it into insufficient. We start with par.2., considering line i_o as unexamined.

If the extreme right element of this series, equalling d_s, was found to be $x_{i_rk_{r-1}}$ (r-one of the numbers 1,2,3,...,s), then all lines, in the chain of which is included the number of column k_{r-1}, are renamed into excessive. If such renamed lines are in lesser number than insufficient examined lines, then we turn to par 4, considering thereat that the renamed lines are subject to checking, i.e. to operations explained in par.4.If the renamed lines are not in lesser number than the insufficient examined, we turn to par.2.,whereby the insufficient examined lines are again added to the non-examined ones.

If there was an equality $N_{ks} = d_s$, and all remaining $N_k = 0$, then after the operations of calculating and adding d_s the problem appears to be solved.

In the presence of $N_k > 0$ all lines, in the chain of which is included the number of column k_s, by renaimng into excessive we follow exactly as in the case, discussed

in previous chapter.

Par.4. Checking excessive lines

We begin the check with one of the lines subject to checking and having the lowest rank. If there are no such lines, we go over to par.2. We designate the number of the checked line by i_o, and its rank - by t.

We establish, whether the checked line i_o has a connection with the column, fed with insufficiently examined line of the rank t - 1, or, at t = 0, with the unsatisfied column. If there is no such column, we determine, whether line i_o has a connection with the column, supplied with insufficiently examined line of rank t. If line i_o appears to be connected with the unsatisfied column, it acquires the chain $i_o k_o$, where k_o - the number of the unsatisfied column[1], if line i_o is connected with the column, fed with insufficient line of rank t - 1 or t, it acquires the chain $i_o k_o i_1 k_1 \ldots i_s k_s$, where k_o - number of corresponding column, and $i_1 k_1 i_2 k_2 \ldots i_s k_s$ - chain of line i_1, feeding column k_o. Line i_o is considered further as an insufficiently examined line. We turn to the beginning of par 4.

If there is no column, satisfying one of the above mentioned conditions, then we mention all these columns, which are connected with line i_o and feed insufficiently examined lines of any rank (if there are such columns). Of the number of insufficiently examined lines, feeding the mentioned columns, we separate one of the lines, having the lowest rank, and rename it into non-investigated. Line i_o is considered as checked and we return to the beginning of par 4.

If there are no columns connected with line i_o and fed with insufficiently examined lines, we return to the beginning of par.4, considering line i_o as checked.

Par.5. Transformation of Matrices

We designate by k' the numbers of dissatisfied columns and columns fed with in

1. The chain, previously given to the line, is considered as having lost the power.

sufficient lines (i.e.numbers of columns, not connected with excessive lines), and by i' - the numbers of excessive lines. For each column k' we determine

$$d_{k'} = \min_{i'} c_{i'k'} - \min_{i} c_{ik'}$$

Assuming that $d = \min_{k'} d_{k'}$. We increase by d the elements of insufficient lines of the matrix $[c_{ik}]$ or, if the number of excessive lines is lower, than the number of excessive , we reduce by d the elements of excessive lines. We notice the changes in the connections between lines and columns, which appear to be the result of changes in the matrix $[c_{ik}]$

In columns k', for which an equality $d_{k'} = d$ took place (we will call such columns critical[1]) new links do appear: if $\bar{k}_p$ - critical column, and i_r - the line for which equality $c_{i_r k_p'} = \min c_{i' k_p'}$ took place, then after transforming the matrix $[c_{ik}]$ line i_r and column k'_p appear to be interconnected.

At the same with respect to the columns connected up to the transformation of matrix $[c_{ik}]$ with insufficient as well as with excessive lines (if such columns do exist), should be noticed a discontinuation of connections with insufficient lines.

Par.6.Additional Operations after Transformation of Matrices

We determine, whether there are critical dissatisfied columns connected with absolutely excessive lines.

If a critical dissatisfied column has been detected (we designate it by k_o) which is connected with absolutely excessive line i_o then we raise the value of corresponding variable $x_{i_o k_o}$ by the smallest of the numbers M_{i_o} and N_k, simultaneously reducing both these numbers by the value of the lowest of them. If $M_{i_o} < N_{k_o}$, then we return to the beginning of par.6. If $M_{i_o} > N_{k_o}$, then all lines, the chain of which includes the number of column k_o, are renamed into excessive (and are considered further as subject to checking). After this we return to the beginning of par 6. If $M_{i_o} = N_{k_o}$, then at a condition, that all remaining $N_k = 0$, the problem is considered

1. The name of critical is preserved by its obtained columns to the completion of operation according to par.6.

as solved; otherwise we follow the same procedure as at $M_{i_0} > N_{k_0}$.

In the absence of critical dissatisfactory columns, connected with absolutely excessive lines, we explain, whether there are critical dissatisfactory columns, connected with excessive lines, for which $M_i = 0$. If such a column has been detected, then each excessive line, connected with it, acquires a chain in form of i_0k_0, where i_0 - number of line, k_0 - number of dissatisfied critical column [2], and we consider these lines further on as insufficiently unexamined lines. We return to the beginning of par.6.

If there are no critical dissatisfied columns, connected with excessive lines, then we examine the critical satisfied columns (if there are such) and notice the insufficient lines, feeding these columns. The notices lines (if such have appeared) are considered further as not-examined. After this, and also in case , if there are no critical satisfied columns,fed with insufficient lines, we act as follows. If there are excessive lines subject to check and their number is smaller than the number of insufficiently examined lines, we resort to par.4. If there are no excessive lines subject to check or they are present, but their number is smaller than the number of insufficiently examined lines, we turn to par 2.,whereby in the latter case the insufficiently examined lines are again added to the non-examined.

III. Reasons for the Algorithm.

We will show, that after the final number of operations the algorithm leads to a solution of the problem under question.

1. Before each matrix transformation $\|c_{ik}\|$ the values x_{ik} satisfy conditions (5)-(8).

By direct examination of the algorithm it is easy to convince oneself in the adherence to conditions (5)-(7) at any given stage of solving the problem, as well as in the fact, that before each change over to par.5 there is not a single column

2.If the given line acquired a chain in preceding operations,then it is considered as having lost the power.

or column fed with insufficient line[1]. It is evident from the latter that the feeding of dissatisfied columns and columns, fed with insufficient lines (i.e. $\sum x_{ik}$, at designations, adopted in par.5), can not be increased on account of the excessive lines, otherwise would be infringed the condition (5). It is evident herefrom, that just as for all insufficient lines $M_i = 0$, and all columns, fed with excessive lines, appears to be satisfactory the fact that $\sum xi'_k$ cannot be increased, i.e.condition (8) is actually maintained before each matrix transformation $[c_{ik}]$

2. After the final number of operations the algorithm leads to such values of variables x_{ik} and matrix elements $[c_{ik}]$ at which the conditions (1), (2) and (5) are maintained.

Examination of par.1-4 and 6 shows, that each time after matrix transformation $[c_{ik}]$ (for which is needed a final number of operations, discussed in par.5),as well as during the initial stage of solving the problem the final number of operations leads either to realization of conditions (1) and (2) (for that is sufficient,that for any given column Nk=0) or to the application of par.5 (i.e.to alternate transformation of matrix $[c_{ik}]$

Consequently,it has to be proven that the number of matrix transforms $[c_{ik}]$ i final[2].

Matrices $[c_{i'_k}]$ and $[c_{i''_k}]$ will be called idempotent, or equality

$$c_{i'_r k_p} = \min_i c_{i' k_p}$$

appears to be a necessary and sufficient condition of equality

$$c_{i''_r k_p} = \min_i c_{i'' k_p}$$

If the matrices are idempotent , then conditions (5)-(8) are satisfied at one and the cery same values x_{ik}.

1. To detect and eliminate such "irregular" columns are brought dow, as can be seen with ease, all operation in par.1-x4 and 6. The result of using par.5 is the appearance of new incorrect columns.
2. Condition (5)is maintained, as mentioned above,at any given stage of algorithm application.

Transformation of matrices $[c_{ik}]$, realizable during the application of algorithm, can not lead to the appearance of a matrix, equally potent to the previously obtained ones. The fact is, if such a transform involves a change in value x_{ik}, and consequently, also a change (increase) in $\sum_{i,k} x_{ik}$, corresponding to conditions (5) - (8), then the transformed matrix cannot be idempotent to any of the previously obtained ones, since on any of the preceding stages of applying the algorithm $\sum_{i,k} x_{ik}$ was smaller, and consequently, the values x_{ik} we different.

It can be shown, that in the case, when matrix transformations $[c_{ik}]$ do not bring about any changes in the value x_{ik}, the appearance of idempotent matrices is also impossible.

At any given matrix transformation $[c_{ik}]$ if even one of its elements would not appear minimum in the corresponding column, it then becomes so, i.e. the equality $c_{i_r k_p} = \min_i c_i k_p$ becomes valid for it. To transform matrices $[c_{ik}]$, causing no changes in the x_{ik} values, is valid the following condition: element of matrix converted as result of such transformation from nonminimal to minimal, may further on again appear minimal but only at least when transformation took place, a transformation connected with the change in values x_{ik}. Hence it is evident, that during transformations, not connected with changes in x_{ik} values, idempotent matrices cannot appear.

The above formulated position in turn can be easily proven, on the basis of the following properties of the discussed algorithm:

a) during matrix $[c_{ik}]$ transformation a new minimal element may appear only in the column, which is not fed with excessive lines;

b) during matrix $[c_{ik}]$ transformation the minimal element may transform into nonminimal only in the column, which is fed with excessive lines;

c) transformations of matrix $[c_{ik}]$ causing no changes in the x_{ik} values, they cannot lead to transformations of any given insufficient line into excessive[1].

1. In this case there is only a conversion of certain excessive lines into insufficient.

To produce proof for position par.2 it remains to mention that the number of non-idempotent matrices at given n and m is finite. Hence it follows, that even the number of matrix transformations is also finite, because they cannot lead to the appearance of idempotent matrices. In this way position p.2 is proven.

Remark. Above c_{ik} designated elements of initial matrix $[c_{ik}]$, as well as elements of transformed matrices. If under c_{ik} we were to understand only elements of initial matrix (numbers, established by the conditions of the problem), and by c^o_{ik} to designate the elements of the matrix, obtained as result of realizing the algorithm, then $[c^o_{ik}] = [c_{ik} + \lambda_i]$, where λ_i - certain actual numbers (determinable by operations of adding and subtracting, discussed in par.5). Condition (5) will be expressed in this case in the following manner

$$x_{i_r k_p} = 0, \text{ если } c_{i_r k_p} + \lambda_{i_r} > \min_i [c_{ik_p} + \lambda_i]. \quad (5a)$$

3. $\sum_{i,k} c_{ik} x_{ik}$ acquires the lowest values, possible when adhering to conditions (1) - (3), in this and only in this case, if condition (5a)[1] is fulfilled.

We will adopt for the system of values x_{ik}, meeting the requirements (1)-(3) and condition (5a), designations x'_{ik}, and for the system of values x_{ik} satisfying only conditions (1) - (3), -designations x''_{ik}. Since there is at least one x''_{ik} value for which the condition (5a) is disrupted, it is apparent, that

$$\sum_{i,k} (c_{ik} + \lambda_i) x'_{ik} = \sum_{i,k} c_{ik} x'_{ik} + \sum_{i,k} \lambda_i x'_{ik} < \quad (9)$$
$$< \sum_{i,k} (c_{ik} + \lambda_i) x''_{ik} = \sum_{i,k} c_{ik} x''_{ik} + \sum_{i,k} \lambda_i x''_{ik}.$$

but by virtue of condition (1)

$$\sum_k x'_{ik} = \sum_k x''_{ik} = a_i. \quad (10)$$

Herefrom

$$\sum_{i,k} \lambda_i x'_{ik} = \sum_{i,k} \lambda_i x''_{ik} = \sum_i \lambda_i a_i. \quad (11)$$

1. Position par.3 can be considered as a result of the theorem about L.V.Kantorovich [9] potentials. The fact is, λ_i appear to be potentials of lines, and the smallest for each column sums $c_{ik} + \lambda_i$ - potentials of corresponding columns.

Fom (9) and (11) we obtain

$$\sum_{i,k} c_{ik} x'_{ik} < \sum_{i,k} c_{ik} \bar{\bar{x}}_{ik}. \qquad (12)$$

Since during the realization of operations, provided with algorithm, the requirement (3) is always maintained, it becomes evident from 2 and 3 positions, that after the final number of operations the algorithm actually leads to solving the problem.

Literature

1. A.N. Tolstoy. Methods of Finding the Smaller Total Kilometrage when Planning Transportation. Collection of Reports Planning of Transportation, Moscow, Transpechat' 1930

2. A N. Tolstoy, Methods of Eliminating Nonrational Transportations when Setting Up Operational Plants. Moscow, Transzheldizdat, 1941

3. L. V. Kantorovich; Mathematical Methods of Organizing and Planning Transportation, Leningrad, izdatel'stvo LGU, 1939

4. L. V. Kantorovich. On Methods of Analyzing of Certain Extreme Planning - Industrial Problems. Doklady Akademii Nauk SSSR, 115, No.3, 1957

5. L. R.Ford and D. R.Fulkerson. A simple algorithm for finding maximal network flows and an application to the Hitchcoock problem. Rand Corporation, Santa Monica, 1955, 743

6. J Muncres. Algorithms for the assignement and transportation problems. Journal of the society for industrial and applied mathematics, 5, N 1, 1957

7. A.Aleksandrov, A.Lurye, Yu.Oleynik. Use of Electronic Computers in Operational Planning. Auto Transportation No. 6, 1959

8. A.L.Lurye; Methods of Attaining Least Range of Loads for Compiling Transportation Schemes. Collection of reports. Application of Mathematics in Economical Investigations. Sotsekgiz, 1959

9. L. V. Kantorovich; On the Application of Masses. Doklady Akademii Nauk SSSR 37, No. 708, 1942

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE	Nr. Copies
HEADQUARTERS USAF	
AFCIN-3D2	1
ARL (ARB)	1
OTHER AGENCIES	
CIA	1
NSA	6
DIA	9
AID	2
OTS	2
AEC	2
PWS	1
NASA	1
ARMY	3
NAVY	3
RAND	1

MAJOR AIR COMMANDS	Nr. Copies
AFSC	
SCFDD	1
ASTIA	25
TDBTL	5
TDBDP	5
SSD (SSF)	2
APGC (PGF)	1
ESD (ESY)	1
RADC (RAY)	1
AFSWC (SWF)	1
AFMTC (MTW)	1